



Sending Email

Link submit:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1927

Solution:

C++	http://ideone.com/cFDTV0
Java	http://ideone.com/pD1wth
Python	http://ideone.com/lR3qVe

Tóm tắt đề:

Có n máy chủ được kết nối bằng cáp mạng. Có m đường cáp mạng, mỗi cáp kết nối hai máy tính với nhau với độ trễ được tính bằng mili giây để gửi một email. Tìm thời gian ngắn nhất để gửi một email từ server S đến server T . Giả sử không có tình trạng mất kết nối giữa hai server.

Input:

Dòng đầu tiên là chứa số N là số lượng bộ test. Mỗi bộ test có định dạng như sau:

- Dòng đầu chứa các số: n, m, S, T ($2 \leq n \leq 20.000$, $0 \leq m \leq 50.000$, $0 \leq S, T < n$, $S \neq T$) lần lượt là số lượng server, số lượng cáp nối và hai server bạn cần tìm kết nối với nhau.
- m dòng tiếp theo mỗi dòng chứa bộ ba số: hai số đầu chỉ số hai server kết nối với nhau (kết nối hai chiều) nằm trong đoạn $[0, n - 1]$, số tiếp theo là w ($0 \leq w \leq 10.000$) là độ trễ của việc kết nối hai server.

Output:

Với mỗi bộ test bạn sẽ in ra "Case #x: " với x là số thứ tự bộ test bắt đầu từ 1, sau đó là độ trễ thấp nhất để kết nối hai server ở bộ test thứ x . Nếu không tìm ra được đường kết nối giữa hai server với nhau thì in ra "unreachable".

Ví dụ:

3 2 1 0 1 0 1 100 3 3 2 0 0 1 100 0 2 200 1 2 50	Case #1: 100 Case #2: 150 Case #3: unreachable
--	--

Giải thích ví dụ:

Ví dụ trên gồm 3 bộ test.

Bộ 1: Có 2 máy chủ và 1 đường kết nối. Cần tìm độ trễ thấp nhất từ máy chủ số "0" sang số "1". Trong bộ này chỉ có duy nhất một đường kết nối từ "0" đến "1" nên kết quả cần tìm là 100.

Bộ 2: Có 3 máy chủ và 3 đường kết nối. Cần tìm độ trễ thấp nhất từ máy chủ số "2" sang số "0". Trong bộ này có 3 đường kết nối, tuy nhiên chúng ta cần tìm đường đi tối ưu là 150.

Bộ 3: Có 2 máy chủ tuy nhiên 2 máy chủ này không có đường kết nối với nhau, vì thế kết quả bạn in ra là "unreachable".

Hướng dẫn giải:

Bài này khá đơn giản, bạn đọc tốt dữ liệu đầu vào, mỗi ví dụ bạn cần tìm đường đi ngắn nhất bằng thuật toán Dijkstra. Mỗi lần xong bạn sẽ in kết quả ra và xóa các tham số cần thiết trong graph, dist.

Độ phức tạp: $O(T * E \log V)$ với T là số lượng bộ test, E là số lượng cạnh (cung) của đồ thị, V là số lượng đỉnh của đồ thị.